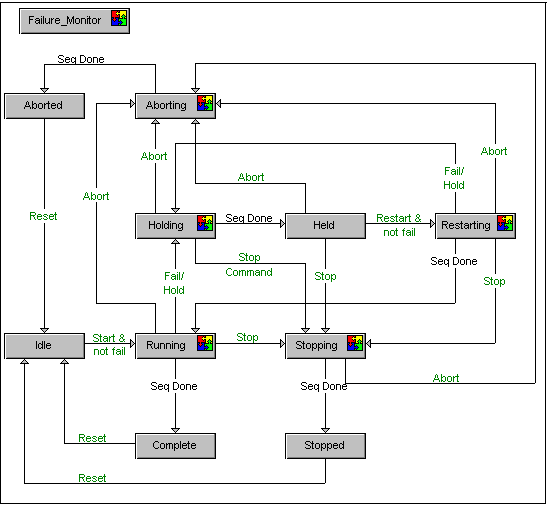
# 一、Deltav 阶段相关

## 状态迁移图：

阶段变成必须和状态转换逻辑一致，如下图所示。途中描述了活动状态（由阶段逻辑支持）、稳态（不需要阶段逻辑）以及两种状态之间的路径。



### 1．运行逻辑

运行逻辑是执行阶段操作的常规逻辑。运行逻辑在阶段受到启动命令时启动，状态进而转为运行中（Running）。当完成运行逻辑时，状态进入完成（COMPLETE）。

当收到启动命令时，阶段逻辑在执行运行逻辑之前，需要阶段下装所有的批量参数。当完成阶段时，阶段逻辑在更改状态为完成之前，需为阶段上传所有的报告参数。

### 2．保持逻辑

保持逻辑用于在检测到故障时，让过程进入安全状态。当收到保持，或者设置了FAIL\_INDEX参数时，阶段会从运行中（RUNNING）或重启中（Restarting）状态转到保持中（Holding）状态，并开始执行保持逻辑。

阶段检查请求是否已由运行逻辑执行，如果是，保持请求，并发送ABORT\_REQUEST代码给批量执行器。

如果状态是运行中（RUNNING），运行逻辑将暂停，直到阶段返回到运行状态。如果之前状态是重启中（RESTARTING），重启逻辑会停止。

当完成保持逻辑，状态更改为已保持（HELD），并且保留已保持（HELD）状态直到用户采取动作。

（不建议在保持逻辑中发送请求。但是，如果必须要发送请求，一定要检查WatchDog参数是否已跳车。若果WatchDog参数跳车，编写不发送请求的逻辑。否则，可能造成阶段无法恢复。）

### 3．重启逻辑

重启逻辑是用于从保持中（HOLDING）状态返回到运行中（RUNNING）状态的逻辑。当收到重启（Restart）命令时，状态转换到重启中（RESTARTING）状态，并且开始执行重启逻辑。当完成重启逻辑时，状态更改为运行中（RUNNING），并且再次执行运行逻辑。使用RESTART\_TYPE参数，可以在进入保持中（Holding）状态或从头开始时，控制是否要在原来的位置继续运行逻辑。

（不能在组态时修改阶段的重启类型。可通过编写RESTART\_TYPE参数自动在阶段中修改。）

### 4．停止逻辑

停止逻辑是用于在收到停止（Stop）命令时，让过程进入安全状态的逻辑。状态更改为停止中（Stopping），所有运行的逻辑都将停止，并且停止逻辑将执行。

当完成停止逻辑时，状态更改为已停止（Stopped），一直到收到重置（Reset）命令。

（不能从已停止状态返回到运行中（Running）状态）。

### 5．中止逻辑

中止逻辑用于使进程达到安全状态。在收到中止（Absort）命令时，状态更改Absorting，运行中的所有逻辑都将停止，并执行中止逻辑。

中止逻辑完成后，状态更改为ABSORTED，并保持ABSORTED直到收到重置（Reset）命令。

（不能从已中止状态返回到运行中（Running）状态）。

### 6．故障监视

Fail\_Monitor复合模块用于监视过程条件，以检测问题。在阶段为活动时，Fail\_Monitor以阶段的组态扫描速率一直运行。

编写代码设置FAIL\_INDEX参数，使阶段进入保持中（HOLDING）状态。首先。FAIL\_INDEX在阶段中运行；其次，当阶段状态不是空闲（Idle）、已完成（Completed）、已停止（Stopped）或者已中止（Absorted）时运行；最后，当正在运行当前状态逻辑时不运行。

缺省情况下，FAIL\_INDEX包含一个嵌入式的、监控过程条件的Calc/Logic功能块。Calc/Logic功能块中所包含的表达式，用于检查Watchdog参数当前值的状态，如果状态为故障（Failed），则设置阶段到已故障（Failed）。

表达式也用于检查Switched\_Over参数的当前值，如果其值为真（True），设置Fail\_Index参数到检测到控制器切换。

如果Watchdog参数状态不是故障，或者Switched\_Over值为False，则没有检测到故障，阶段继续运行。

*以下是Fail\_Monitor复合模块的缺省Calc/Logic功能块表达式。*

*if ( '/+/WDOG\_STATE.CV' = '$phase\_wdog\_states:FAILED' ) then*

*'/+/FAIL\_INDEX' := 'phase\_failures:PLM Watchdog Failed';*

*endif;*

*if ( '/+/SWITCHED\_OVER.CV' = TRUE ) then*

*'/+/FAIL\_INDEX' := 'phase\_failures:Controller Switchover Detected';*

*'/+/SWITCHED\_OVER.CV' := FALSE;*

*endif;*

（可以编辑Fail\_Monitor复合模块里的嵌入式Calc/Logic功能块来更改如上的表达式）

## 阶段逻辑参数：

### 1．AUTO\_RESTART：BOOL

用来说明阶段是否从一个保持中（Holding）条件自动重启。

如果阶段处于已保持（Held）状态，并且AUTO\_RESTART设为真，阶段立即执行其REATART\_LOGIC，并且返回到运行中（RUNNING）状态（操作员不用按重启）.

当阶段仍处于Deltav批量控制下时，操作员可从Deltav Operator重启阶段。

单元模块也提供AUTO\_RESTART参数。设置时，它向下传递给单元模块中的所有阶段。

缺省值为假。

阶段必须处于保持中（Holding）或者已保持（Held）状态，以设置参数为真。

### 2．BATCH\_ID ：字符串

在运行时间为阶段记录批量的ID。

### 3．BCOMMAND：16位无符号整型

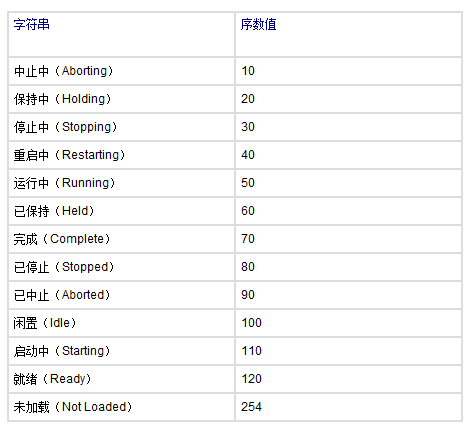
命令阶段进入某个状态，如状态迁移图定义。该参数只用于为Deltav批量执行器。阶段的所有者是Deltav批量。

（不要直接写入该参数，否则会造成批量执行器失去对阶段的控制。）

### 4．BSTATUS：命名集

显示阶段的状态。

该参数使用phase\_state命名集。



*（就绪、未加载 Batch无此概念）*

### 5．CLASS：字符串

返回阶段的阶段类名。

### 6．COMMANDMASK：16位无符号整型

只用于显示用户可以发送的有效命令组。

### 7．CONFIRM\_FAIL：BOOL

激活复合功能故障的确认。默认为假。

### 8．DOWNLOAD\_REQ：BOOL

表示下装（DownLoad）命令已收到，但仍未确认。在检测或动作后会有阶段逻辑复位。

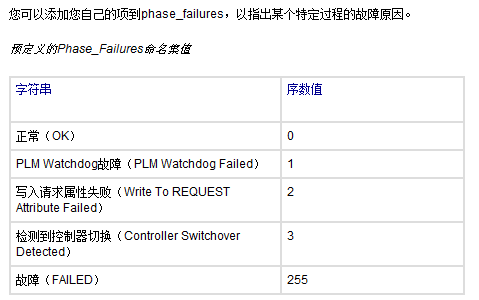
### 9．FAIL\_INDEX：16位无符号整型

显示和定义阶段或阶段界面的故障。其中0=无故障。而非0整数=故障。

范围 0~25。

值255在PHASE\_FAILURE命名集中缺省为故障。建议不要将值255写入FAIL\_INDEX

任何小于255的正整数的转换字符串都可以在PHASE\_FAILUERS命名集中指定。



### 10．INITIAL\_STATE：命名集

在下装时定义SFC的状态（顺序激活或者顺序闲置）。确定SFC的初始状态。

在阶段中，它是运行逻辑开始的状态。缺省为顺序激活。

进入复合功能块后，该参数可用。

该参数使用sfc\_initial\_states命名集。

### 11．OWNER：命名集

识别控制模式是否为DeltaV批量，或控制是否为外部的。 当批量在外部运行时，控制阶段迁移条件参数是XCOMMAND，它允许用户控制阶段。

该参数使用phase\_owner\_id命名集。



### 12．PAUSE\_REQ：BOOL

表示操作员请求暂停（Paused）状态。它基于操作员暂停和恢复（Pause and Resume）命令以及单步（Single-step）模式，由批量执行器(Batch Executive)设置和清除。

### 13．PAUSED：BOOL

显示阶段处于暂停状态。它基于PAUSE\_REQ标识，由阶段逻辑设置和清除。

### 14．PHASE：字符串

返回操作员用于阶段的阶段缩略名。缩略名在单元中组态。

### 15．PHASE\_ACT：BOOL

确定报警的产生是否跟单元上的阶段有关。  
在阶段逻辑的每次执行中计算故障条件，用于在ALARMENB为真时设置PHASE\_ACT。

FailureCondition = TooLong || (FAILINDEX <> 0)|| (PHASE\_ENABLE==FALSE) 其中TooLong = (TOOLONGTIME > 0) && (TIME > TOOLONGTIME) 并且TIME是阶段已经执行的时间。

PHASE\_ACT参数由在模块级别的相关模块报警参数引用。

### 16．PHASE\_CURRENT：BOOL

指出加载的阶段是否正在使用阶段定义的最新版本。一般来说，如果您下装新的阶段定义到一个控制器，而该控制器带有使用较早定义的，已经载入的阶段，阶段会继续使用较早的定义，直到它重新加载。

如果加载的阶段使用最新的定义，则PHASE\_CURRENT设置为真。

如果加载的阶段使用较早的定义，则PHASE\_CURRENT设置为假。

该参数为只读。

### 17．PHASE\_ENABLE：BOOL

使能阶段逻辑执行。真＝使能；假＝禁用。

### 18．PROMPT\_BOOL：BOOL

用于储存操作员对布尔提示的答复（操作员可以选择是或否作为答复）。

### 19．PROMPT\_FLOAT：浮点数

用于储存操作员对浮点数提示的答复（操作员可以在空格中输入一个实数作为答复）。

### 20．PROMPT\_INT：32位有符号整型

用于储存操作员对整数提示的答复（操作员可以在空格中输入一个值作为答复）。

### 21．PROMPT\_STRING：字符串

用于储存操作员对字符串提示的答复（操作员可以在空格中输入一个字符串作为答复）。

### 22．REQDATA1…5：16位无符号整型

用于在阶段逻辑和DeltaV批量执行器（Batch Executive）之间传递请求数据。

### 23．REQUEST：16位无符号整型

来自阶段逻辑发送给DeltaV批量执行器的请求。

### 24．RESTART\_TYPE：命名集

在重启时指定RUN\_LOGIC是否应该在SFC的第一步恢复，或在处于已保持（ HELD）的SFC步骤恢复。

该参数使用phase\_restart\_types命名集。



### 25．SINGLE\_STEP：BOOL

显示连续暂停请求。

### 26．SIZE：32位无符号整型

包括实例化阶段的估计大小。当创建阶段以确定是否有足够的可用存储空间启动阶段时，使用这个参数。

### 27．STATE\_TIME：浮点数

显示模块保持在当前状态的时间。

### 28．STEP\_INDEX：16位无符号整型

经用户组态，STEP\_INDEX可用于指出阶段的每个步骤以跟踪运行的阶段。它是由阶段逻辑以整数写入的，由配方逻辑或DeltaV Operate使用。

### 29．SWITCHED\_OVER：BOOL

显示正在运行的阶段控制器已经经历了冗余切换。在使用之后，由阶段逻辑清除。

### 30．TOO\_LONG\_TIME：32位无符号整型

设置阶段报警器后的时间。

初始值是在创建阶段时确定的。然后，它可以由用户输入。0值禁用过久的定时，归因于设置阶段报警器。

### 31．UINT：16位无符号整型

显示单元的设备ID。

### 32．WATCHDOG：16位无符号整型

用于检验DeltaV批量执行器的通讯。

当阶段状态不是闲置，且所有者是DeltaV批量时，批量执行器会写入到WATCHDOG参数。阶段逻辑会在每次执行时检查WATCHDOG参数并将其清除。

当WATCHDOG参数是基于单元时（对于单元模块），单元的WATCHDOG参数会为该单元上的活动步骤控制WATCHDOG参数。也就是说，单元阶段的WATCHDOG参数会继承单元的WATCHDOG参数值。

### 33．WDOG\_STATE：命名集

用于表示阶段逻辑和批量执行器之间通讯的状态。

有三种状态：正常（OK）、怀疑（Suspect）和故障（Failed）。

如果WATCHDOG参数已经清除，则状态为正常。如果在检查之后，WATCHDOG参数没有清除，则状态为怀疑。如果经过超时值之后，WATCHDOG参数没有被清除，则状态为故障。

该参数使用phase\_wdog\_state命名集。



### 34．WDOG\_TIME：

WATCHDOG参数最后被清除时开始记录时间。

### 35．XCOMMAND：命名集

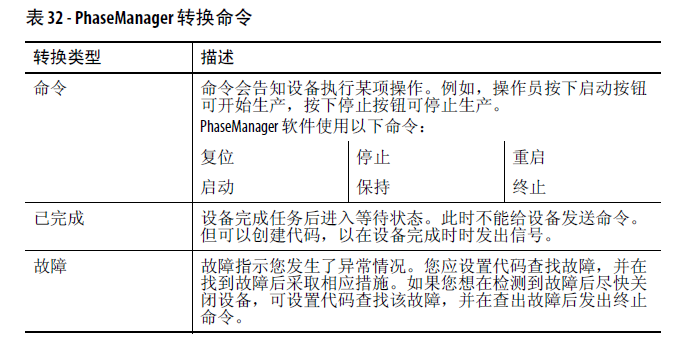
命令阶段进入某个状态，如状态迁移条件图所定义。该参数用于外部运行批量。参见OWNER参数。

该参数使用phase\_command命名集。



# 二、Logic-PhaseManager

## 转换命令



## 指令



## 状态说明

